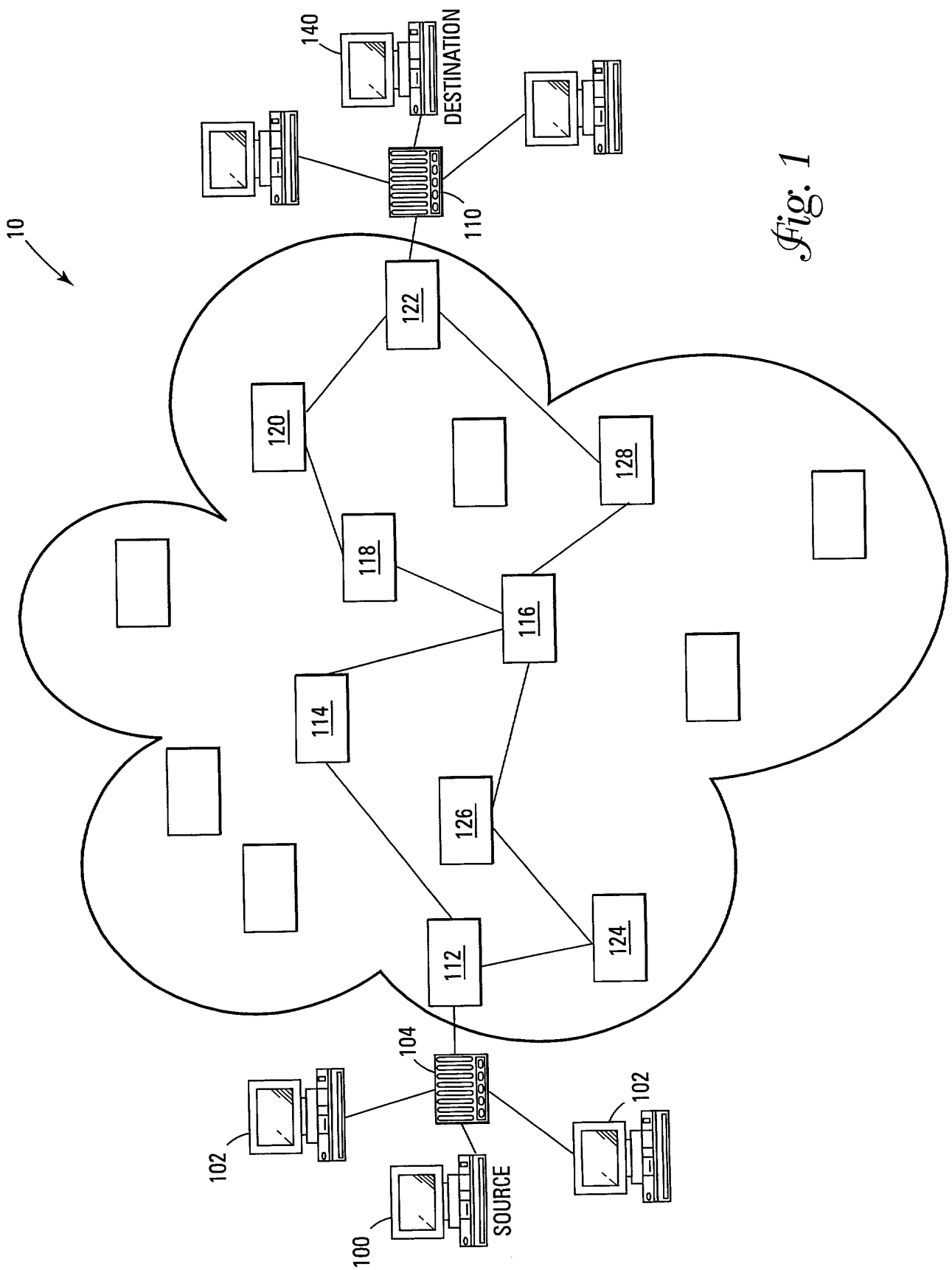
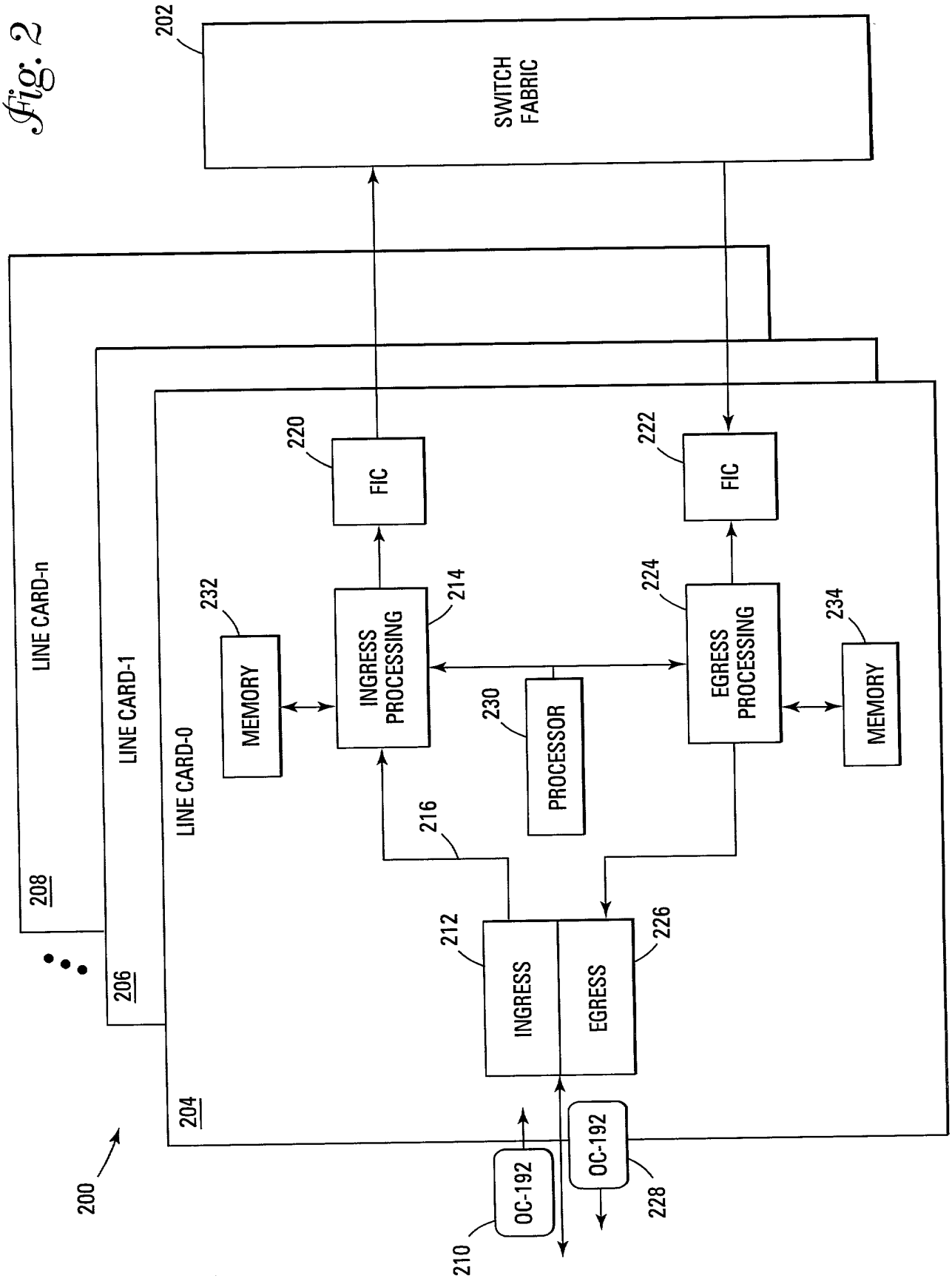

10
140
DESTINATION
110
122
120
128
118
116
114
126
112
124
104
102
102
SOURCE
100

Fig. 1

Fig. 2


202
SWITCH FABRIC
LINE CARD-n
LINE CARD-1
LINE CARD-0
208
206
204
200
220
FIC
222
FIC
232
MEMORY
INGRESS PROCESSING
214
224
EGRESS PROCESSING
234
MEMORY
230
PROCESSOR
216
212
INGRESS
EGRESS
226
OC-192
OC-192
210
228

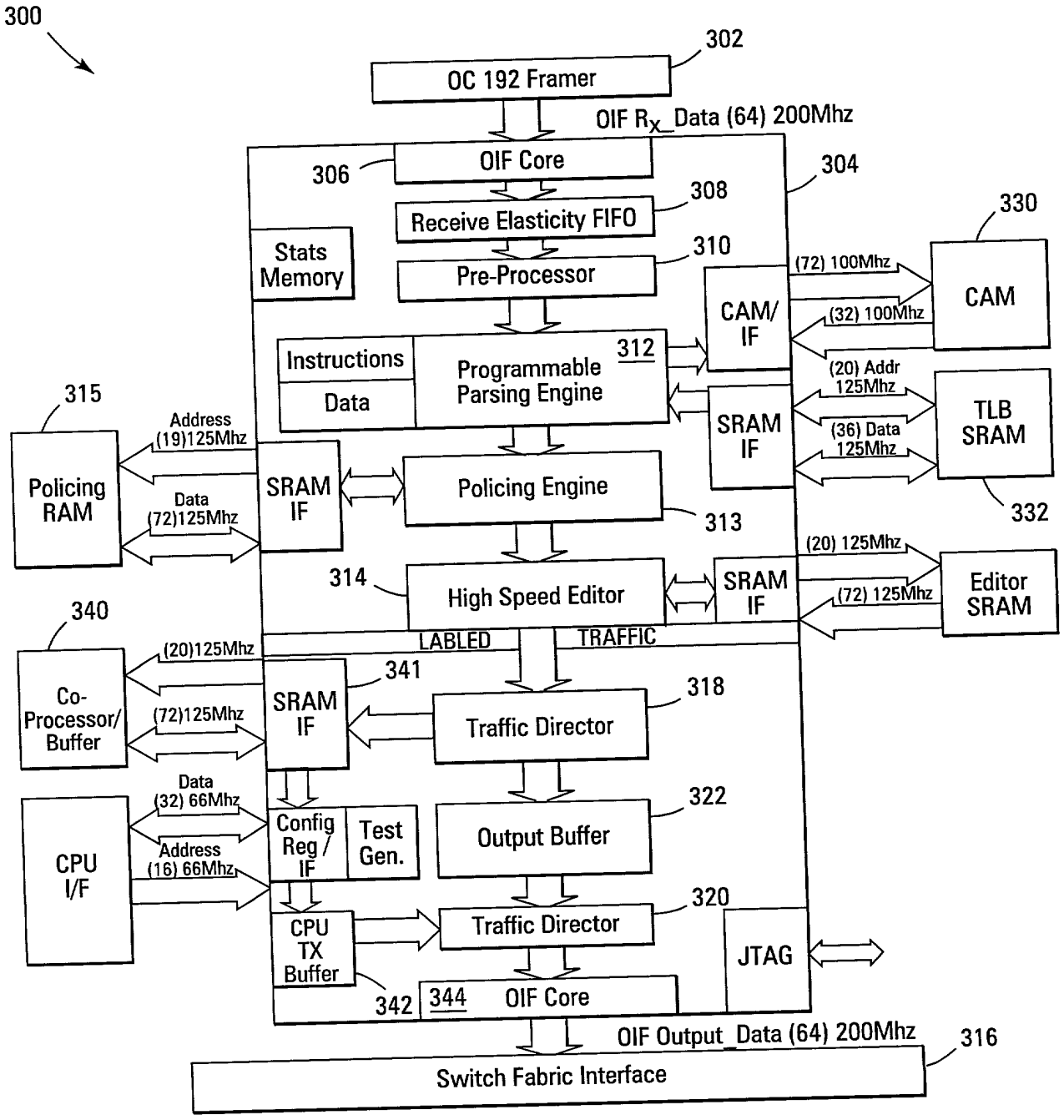

300
302
OC 192 Framer
OIF Rx_Data (64) 200Mhz
306
OIF Core
304
308
Receive Elasticity FIFO
330
Stats Memory
310
Pre-Processor
(72) 100Mhz
(32) 100Mhz
CAM
CAM/ IF
Instructions
Data
Programmable Parsing Engine
312
(20) Addr 125Mhz
(36) Data 125Mhz
TLB SRAM
SRAM IF
315
Address (19)125Mhz
Data (72)125Mhz
Policing RAM
SRAM IF
Policing Engine
313
332
(20) 125Mhz
(72) 125Mhz
Editor SRAM
314
High Speed Editor
SRAM IF
340
LABLED
TRAFFIC
(20)125Mhz
(72)125Mhz
Co-Processor/ Buffer
341
SRAM IF
318
Traffic Director
Data (32) 66Mhz
Address (16) 66Mhz
CPU I/F
Config Reg / IF
Test Gen.
322
Output Buffer
320
CPU TX Buffer
Traffic Director
JTAG
342
344
OIF Core
OIF Output_Data (64) 200Mhz
316
Switch Fabric Interface

Fig. 3

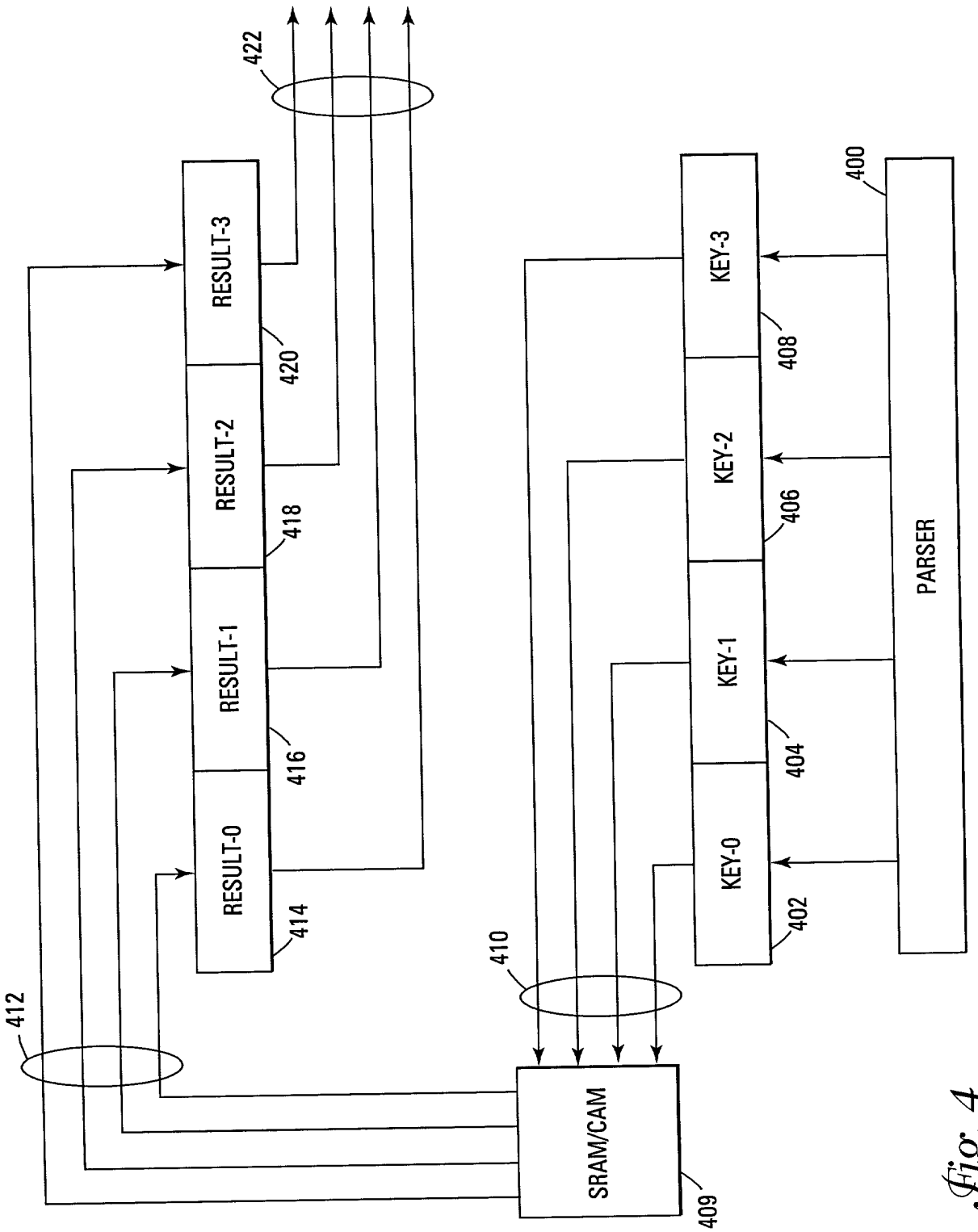

PARSER
400
KEY-3
KEY-2
KEY-1
KEY-0
408
406
404
402
410
SRAM/CAM
409
412
RESULT-3
RESULT-2
RESULT-1
RESULT-0
420
418
416
414
422

Fig. 4

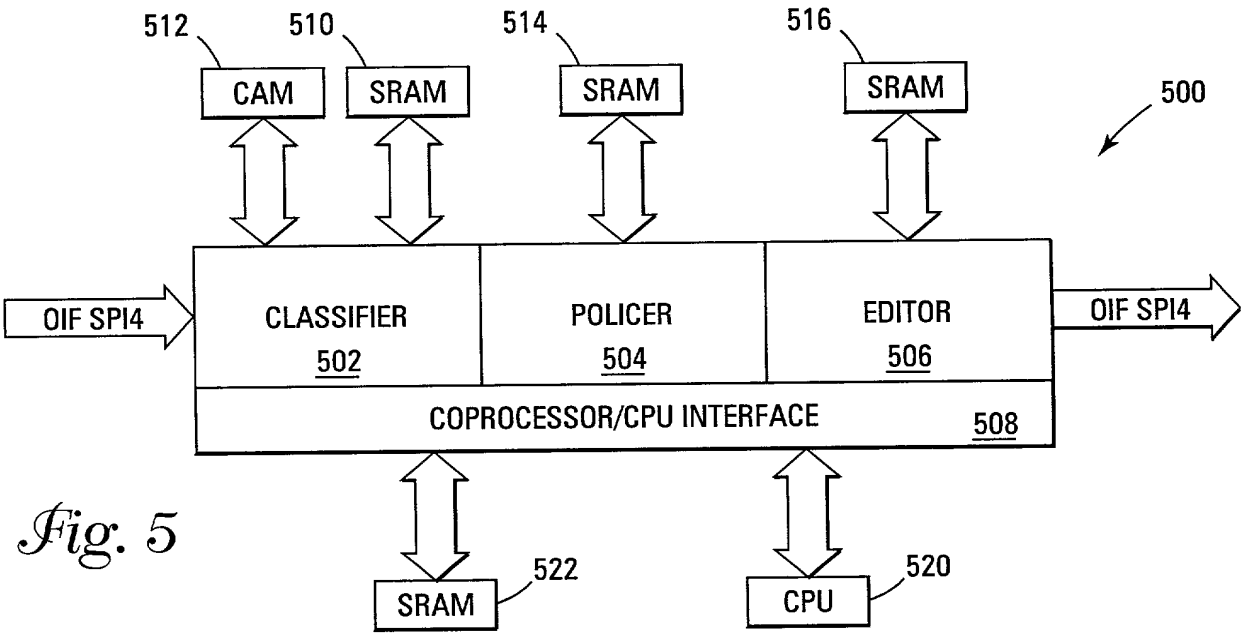

512
510
514
516
CAM
SRAM
SRAM
SRAM
500
OIF SPI4
CLASSIFIER
502
POLICER
504
EDITOR
506
OIF SPI4
COPROCESSOR/CPU INTERFACE
508
SRAM
522
CPU
520

Fig. 5

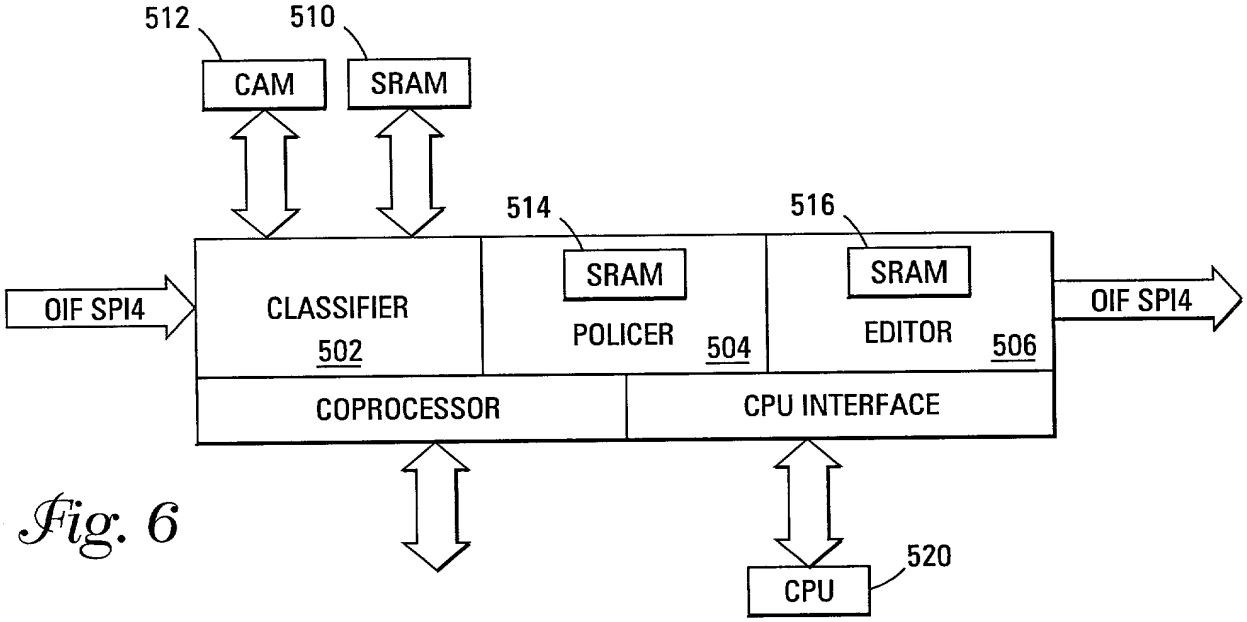

512
510
CAM
SRAM
514
516
SRAM
SRAM
OIF SPI4
CLASSIFIER
502
POLICER
504
EDITOR
506
OIF SPI4
COPROCESSOR
CPU INTERFACE
CPU
520

Fig. 6

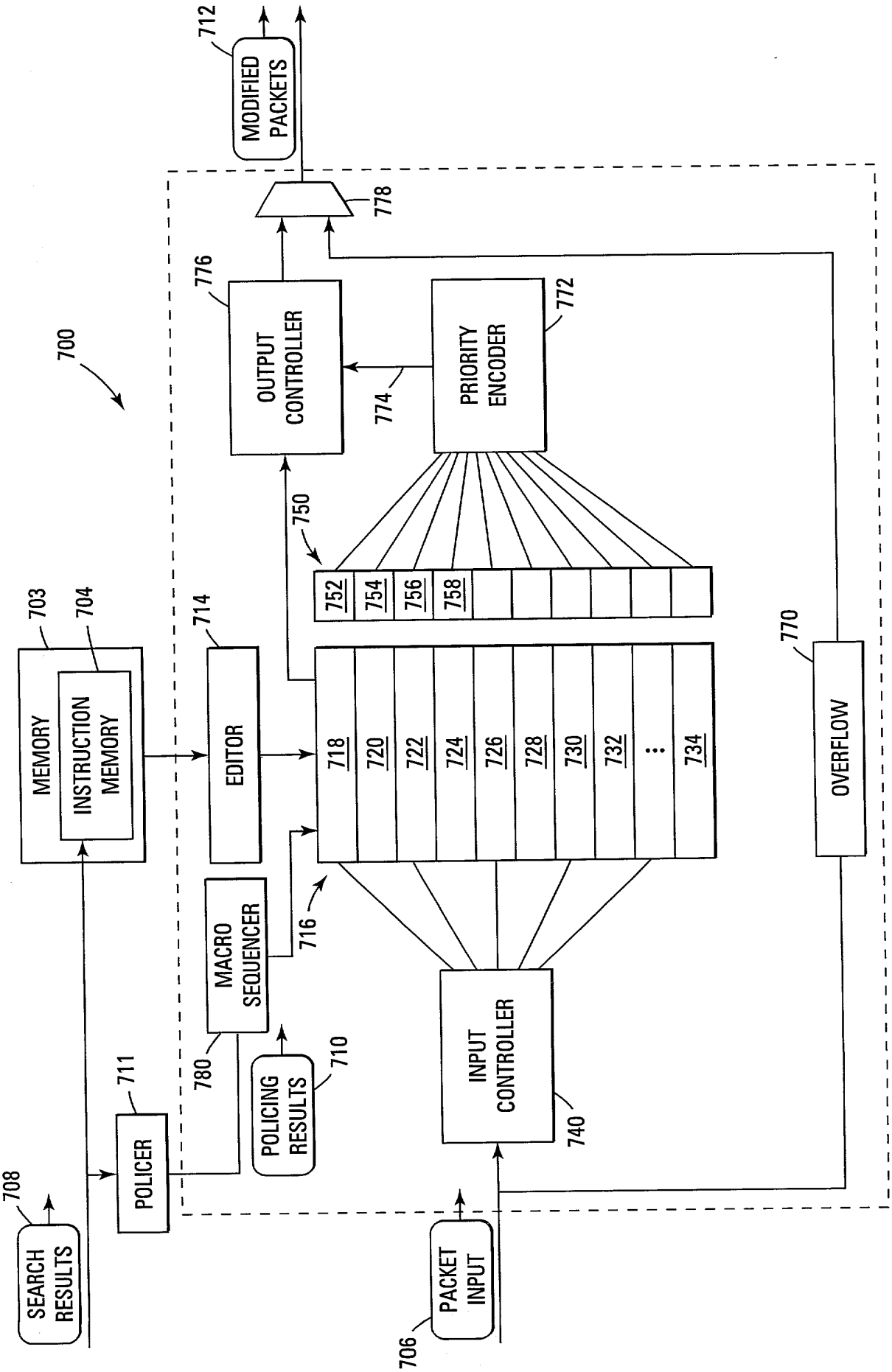

700
MODIFIED PACKETS
712
778
OUTPUT CONTROLLER
776
PRIORITY ENCODER
772
774
750
752
754
756
758
MEMORY
703
INSTRUCTION MEMORY
704
EDITOR
714
718
720
722
724
726
728
730
732
734
OVERFLOW
770
MACRO SEQUENCER
716
780
POLICING RESULTS
710
POLICER
711
INPUT CONTROLLER
740
SEARCH RESULTS
708
PACKET INPUT
706

Fig. 7

T0600T "h086h860

800 802

Editor Instruction	Description
No-Op	No operation is performed
Write1	General purpose single word write
Write2	General purpose double word write
Delete1	General purpose single word delete
Delete2	General purpose double word delete
Read-Modify-Write with Mask	Single word modify; mask provided with data
Read-Modify-Write with Default Mask	Mask provided from one of a plurality of default masks
Swap	Swap the current top MPLS label
Swap/Push1	Swap the current top MPLS label and push one other to the top
Push1	Push one MPLS label to the top of the MPLS stack
Swap/Push2	Swap the current top MPLS label and push two others to the top
Push2	Push two MPLS labels to the top of the MPLS stack
Pop1/Swap	Pop the current top MPLS label and swap the next
Pop1	Pop the current top MPLS label
Pop2	Pop the two current top MPLS labels
PopAll	Pop all MPLS labels from stack

Fig. 8

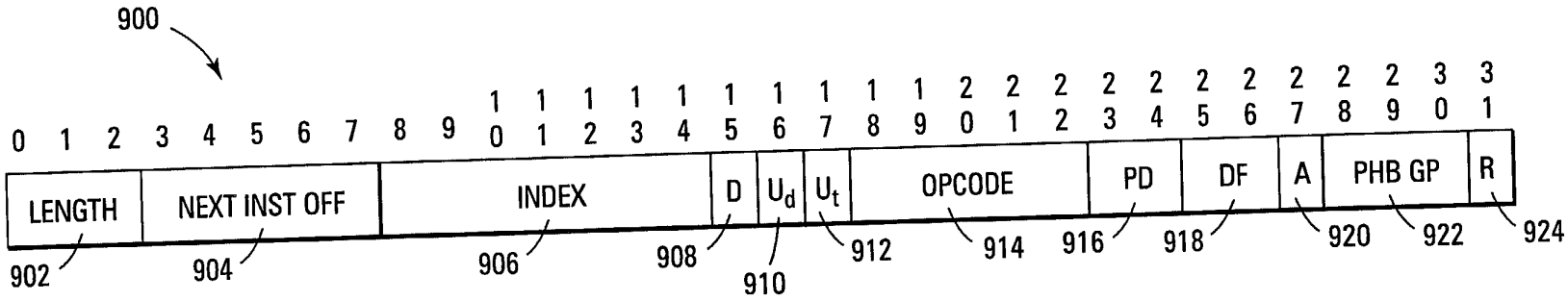

900
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
LENGTH
NEXT INST OFF
INDEX
D
Ud
Ut
OPCODE
PD
DF
A
PHB GP
R
902
904
906
908
910
912
914
916
918
920
922
924

Fig. 9

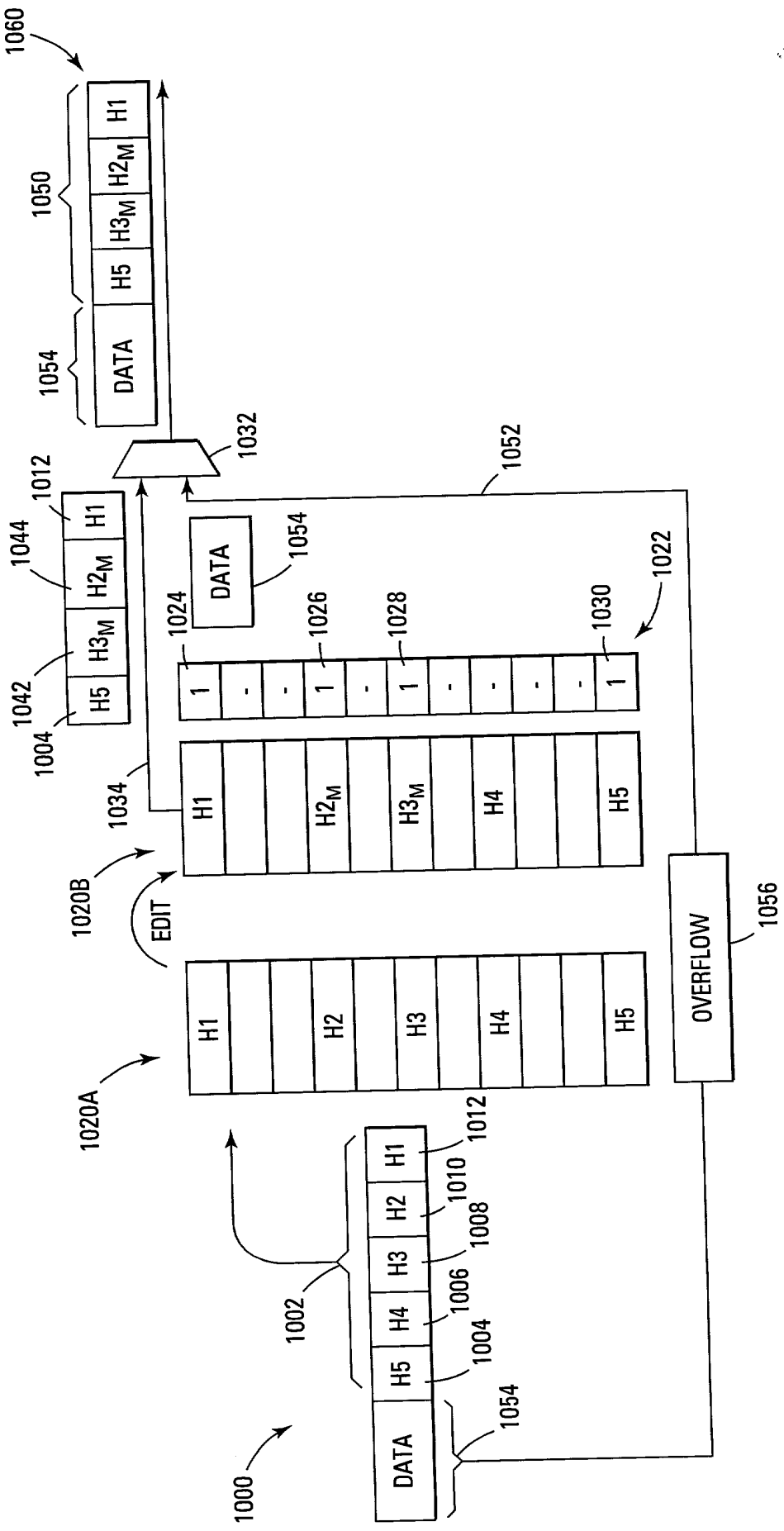

1060
1050
1054
DATA
H5
H3M
H2M
H1
1032
1052
1012
1044
1042
1004
1034
1020B
EDIT
1020A
1000
1002
1022
1024
1026
1028
1030
1054
1056
OVERFLOW
1010
1008
1006
H2
H3
H4
H5

Fig. 10

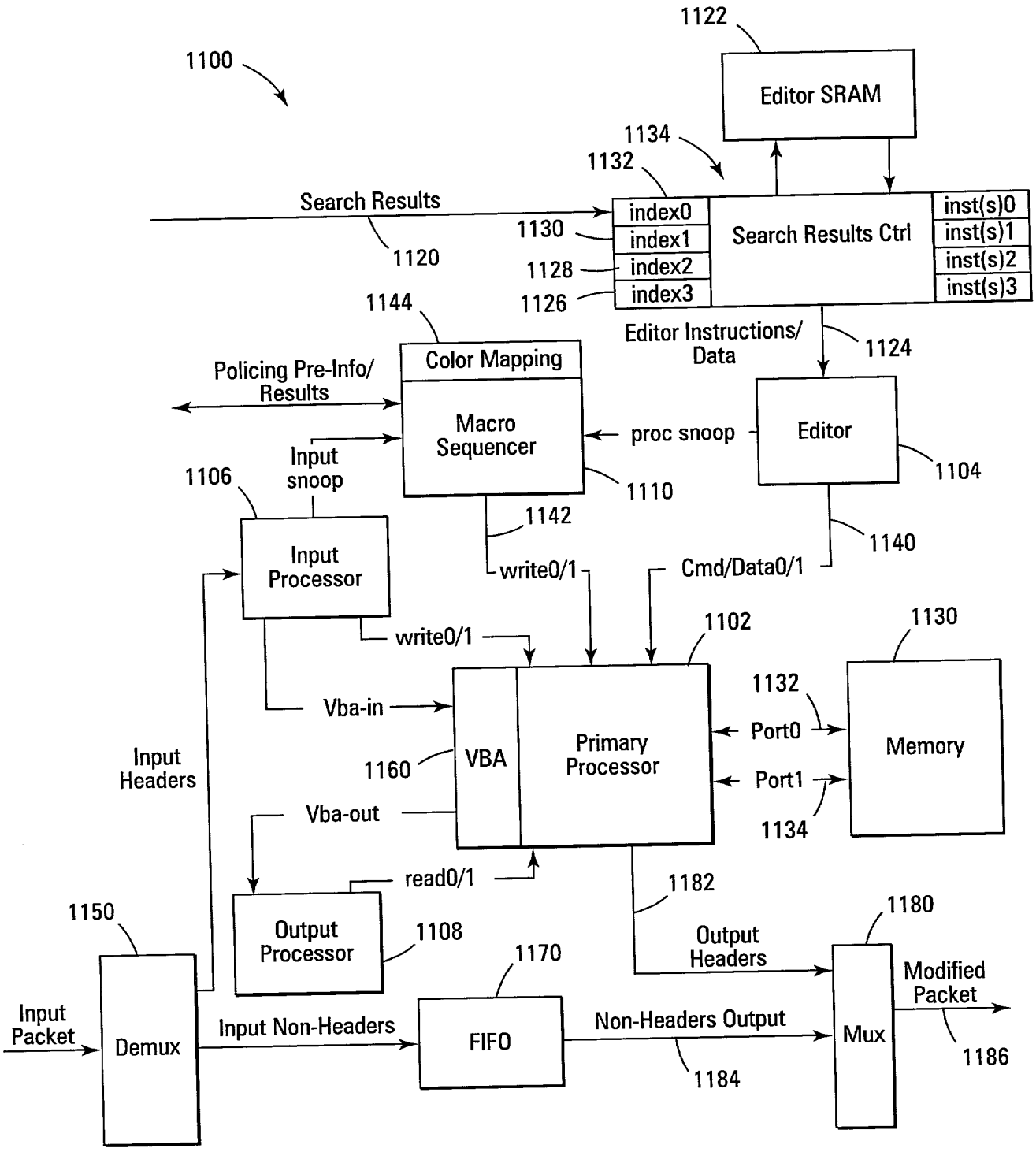

1100
1122
Editor SRAM
1134
1132
Search Results
1120
index0
index1
index2
index3
1130
1128
1126
Search Results Ctrl
inst(s)0
inst(s)1
inst(s)2
inst(s)3
1144
Color Mapping
Editor Instructions/
Data
1124
Policing Pre-Info/
Results
Macro
Sequencer
proc snoop
Editor
1104
Input
snoop
1106
1110
1142
1140
Input
Processor
write0/1
Cmd/Data0/1
write0/1
1102
1130
Vba-in
1132
Port0
Memory
Input
Headers
1160
VBA
Primary
Processor
Port1
1134
Vba-out
read0/1
1182
1150
1180
Output
Processor
1108
Output
Headers
1170
Modified
Packet
Input
Packet
Demux
Input Non-Headers
FIFO
Non-Headers Output
Mux
1186
1184

Fig. 11

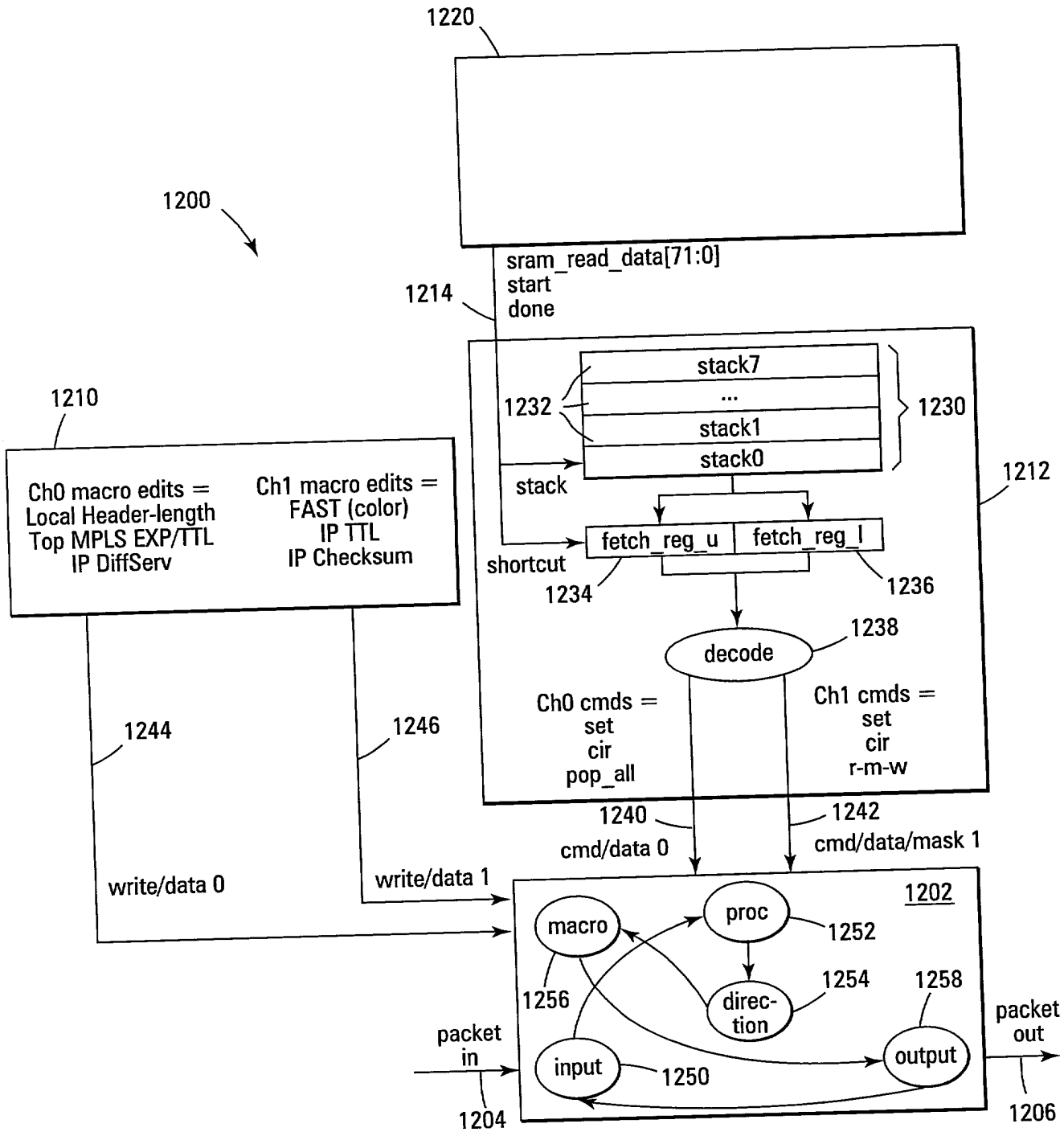

1220
1200
sram_read_data[71:0]
start
done
1214
1210
Ch0 macro edits =
Local Header-length
Top MPLS EXP/TTL
IP DiffServ
Ch1 macro edits =
FAST (color)
IP TTL
IP Checksum
stack7
...
stack1
stack0
1232
1230
stack
1212
fetch_reg_u
fetch_reg_l
shortcut
1234
1236
decode
1238
Ch0 cmds =
set
cir
pop_all
Ch1 cmds =
set
cir
r-m-w
1244
1246
1240
1242
cmd/data 0
cmd/data/mask 1
write/data 0
write/data 1
1202
macro
proc
1252
direc-
tion
1254
1256
1258
packet
in
input
1250
output
packet
out
1204
1206

Fig. 12

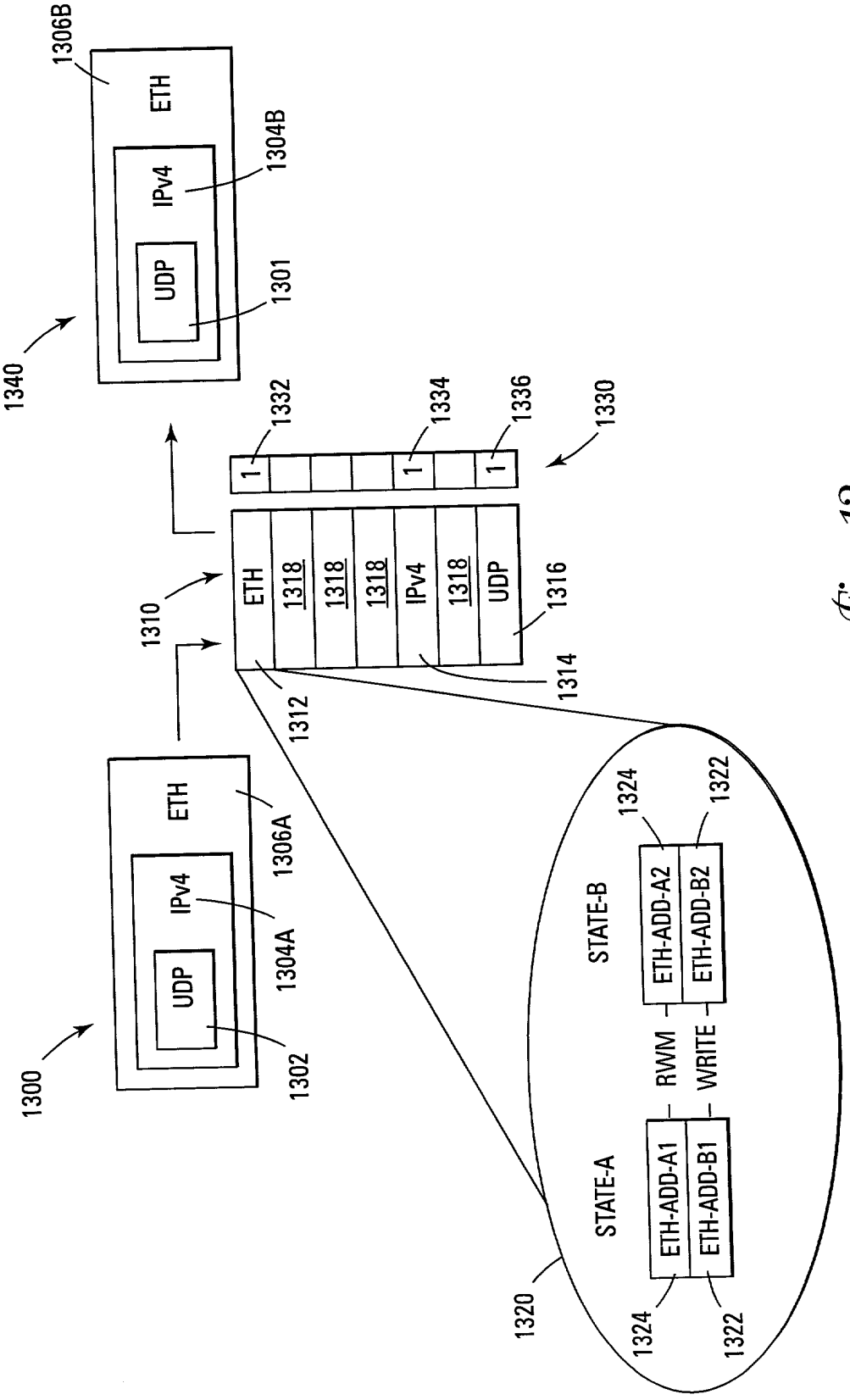

1306B
ETH
IPv4
1304B
UDP
1301
1340
1332
1334
1336
1330
1
1
1
1310
ETH
1318
1318
1318
IPv4
1318
UDP
1316
1314
1312
1306A
ETH
IPv4
1304A
UDP
1302
1300
1324
1322
STATE-B
ETH-ADD-A2
ETH-ADD-B2
RWM
WRITE
STATE-A
ETH-ADD-A1
ETH-ADD-B1
1320
1324
1322

Fig. 13

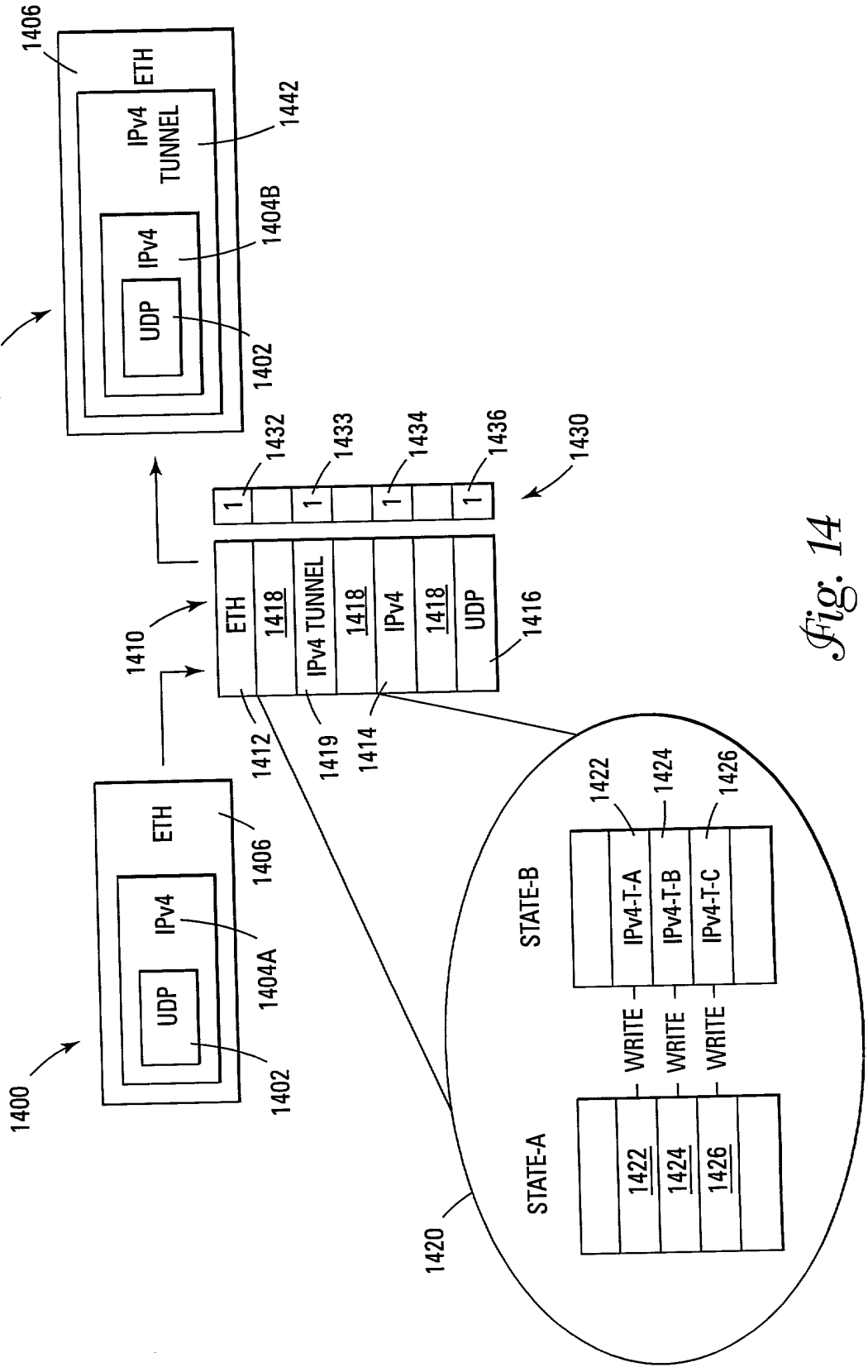

1400
UDP
IPv4
ETH
1402
1404A
1406
1440
1442
1404B
1410
1412
1419
1414
1416
1418
1420
STATE-A
STATE-B
1422
1424
1426
WRITE
1430
1432
1433
1434
1436

Fig. 14

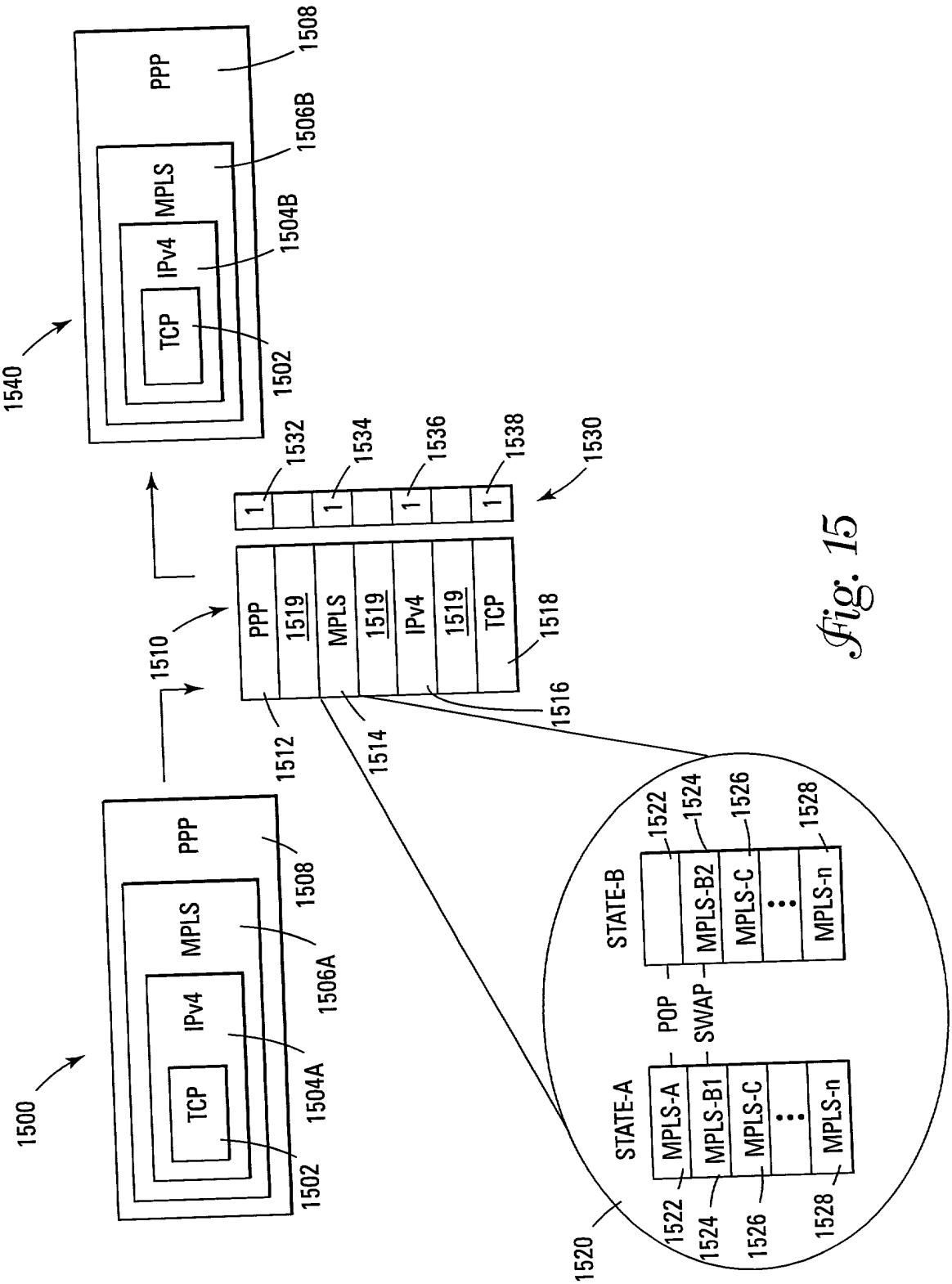

1500
PPP
MPLS
IPv4
TCP
1502
1504A
1506A
1508
1510
1512
1514
1516
1518
PPP
1519
MPLS
1519
IPv4
1519
TCP
1530
1532
1534
1536
1538
1540
PPP
MPLS
IPv4
TCP
1502
1504B
1506B
1508
1520
STATE-A
MPLS-A
MPLS-B1
MPLS-C
MPLS-n
POP
SWAP
STATE-B
MPLS-B2
MPLS-C
MPLS-n
1522
1524
1526
1528

Fig. 15

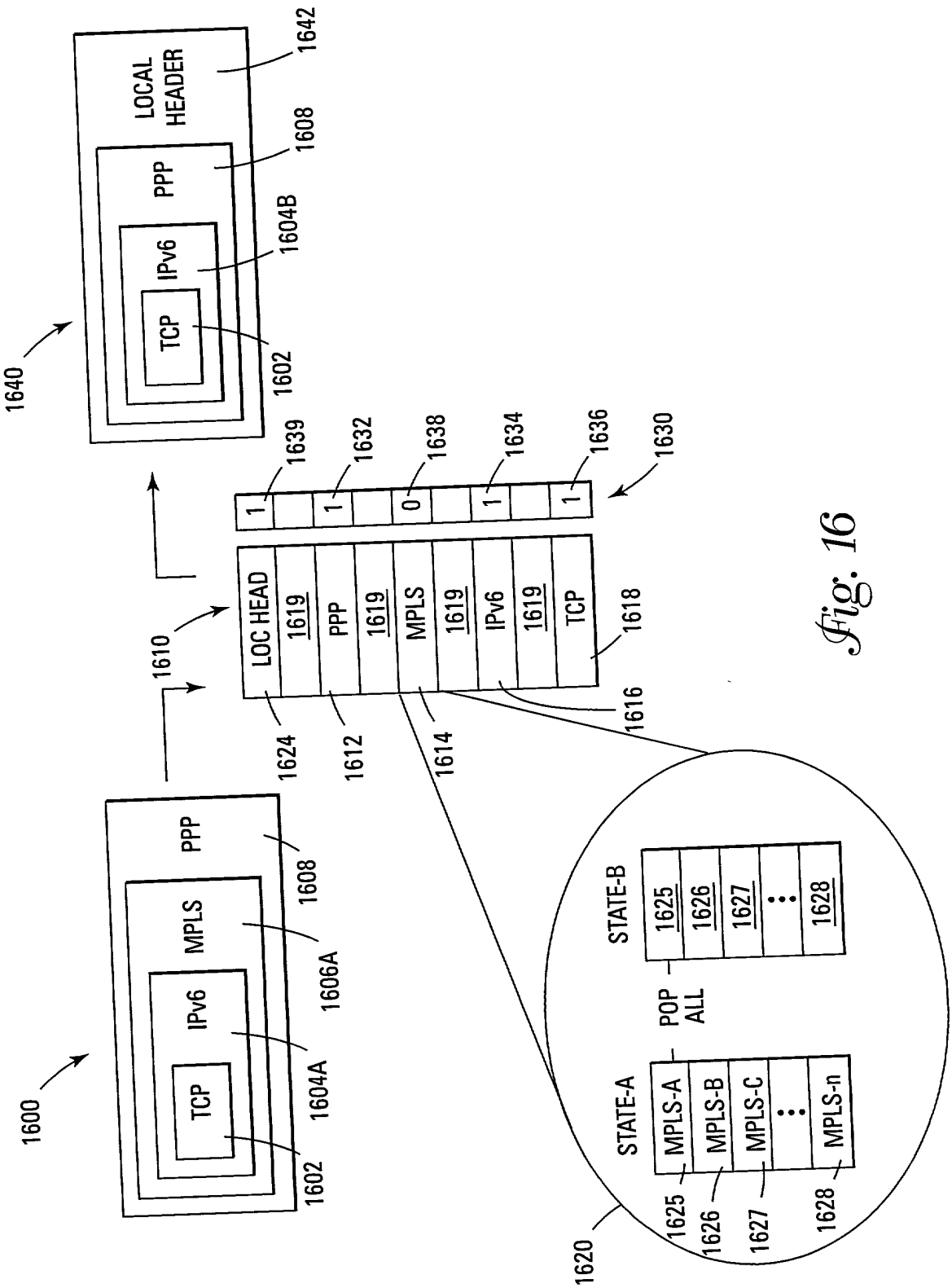

1600
PPP
MPLS
IPv6
TCP
1602
1604A
1606A
1608
1610
LOC HEAD
1619
PPP
1619
MPLS
1619
IPv6
1619
TCP
1618
1616
1624
1612
1614
1630
1639
1632
1638
1634
1636
1640
LOCAL HEADER
1642
PPP
1608
IPv6
1604B
TCP
1602
1620
STATE-A
MPLS-A
MPLS-B
MPLS-C
MPLS-n
1625
1626
1627
1628
POP ALL
STATE-B
1625
1626
1627
1628

Fig. 16

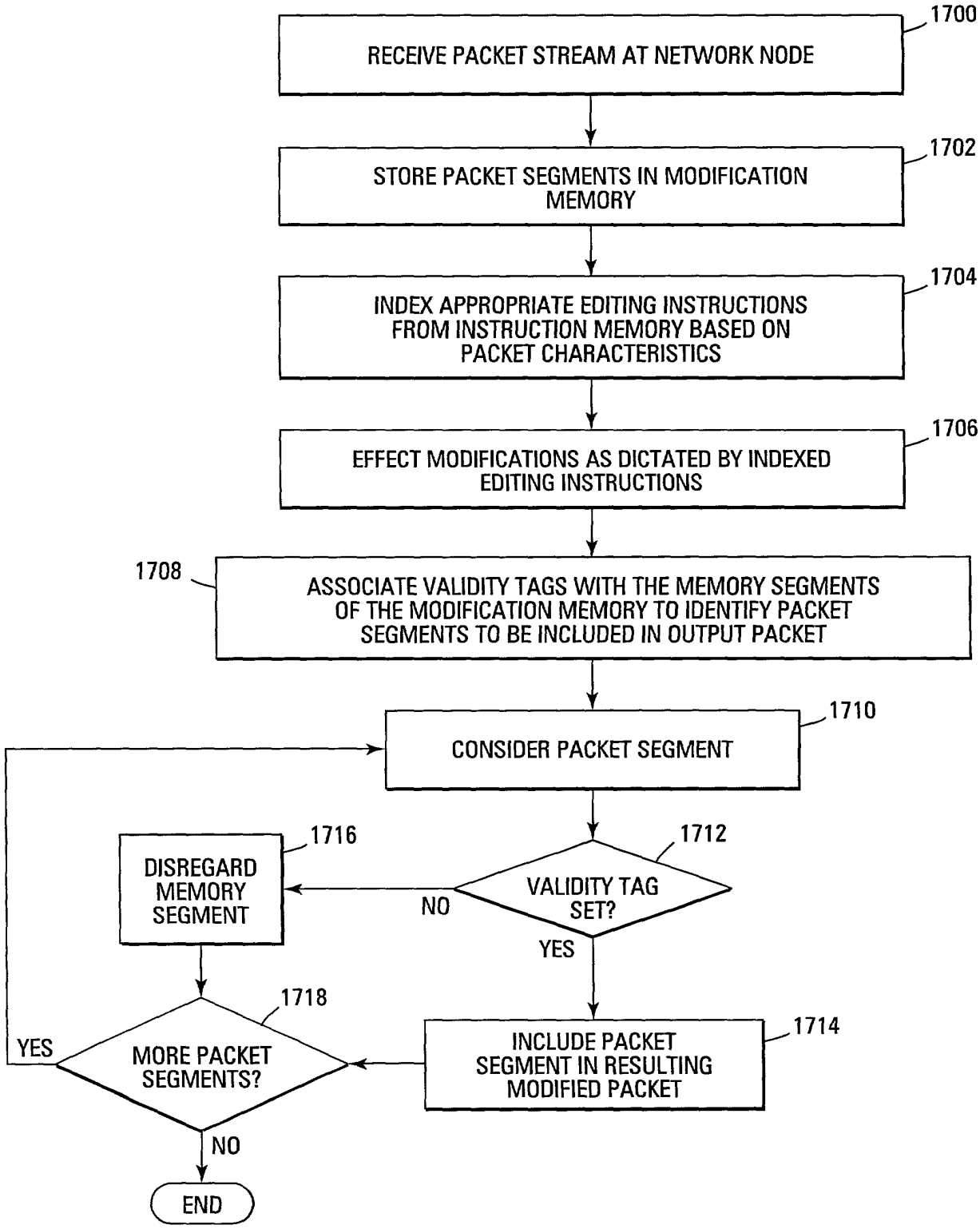

1700
RECEIVE PACKET STREAM AT NETWORK NODE
1702
STORE PACKET SEGMENTS IN MODIFICATION MEMORY
1704
INDEX APPROPRIATE EDITING INSTRUCTIONS FROM INSTRUCTION MEMORY BASED ON PACKET CHARACTERISTICS
1706
EFFECT MODIFICATIONS AS DICTATED BY INDEXED EDITING INSTRUCTIONS
1708
ASSOCIATE VALIDITY TAGS WITH THE MEMORY SEGMENTS OF THE MODIFICATION MEMORY TO IDENTIFY PACKET SEGMENTS TO BE INCLUDED IN OUTPUT PACKET
1710
CONSIDER PACKET SEGMENT
1712
VALIDITY TAG SET?
NO
YES
1716
DISREGARD MEMORY SEGMENT
1714
INCLUDE PACKET SEGMENT IN RESULTING MODIFIED PACKET
1718
MORE PACKET SEGMENTS?
YES
NO
END

Fig. 17